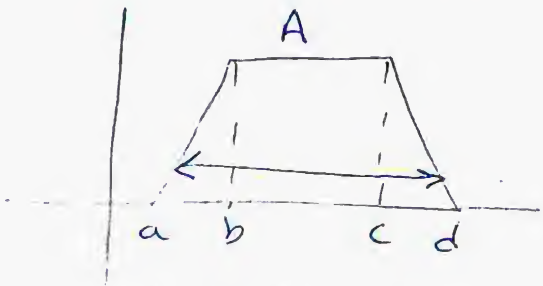
Concepts about fuzzy sets

support

elements of fuzzy set where its MF degree $\neq 0$

ex

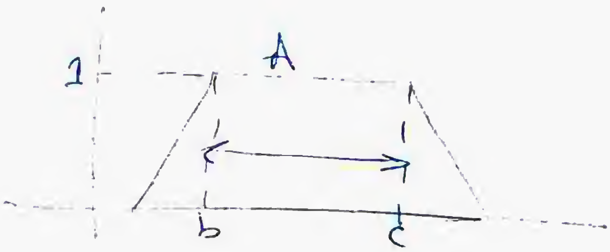
support =]a, d[



Core

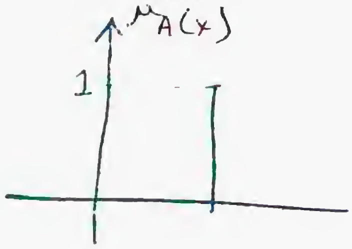
→ elements of fuzzy set where its MF degree = 1

Core(A) = [b, c]



single ton

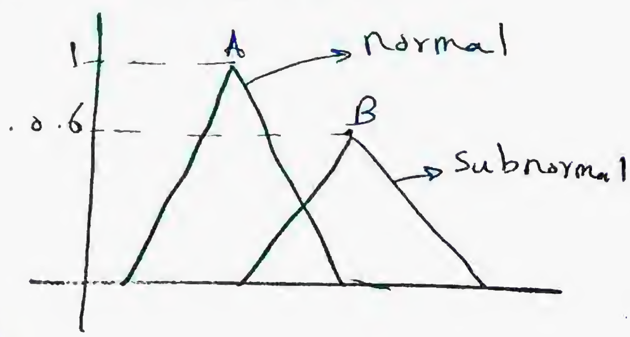
when
→ no. of elements of fuzzy set is equal to 1 with $\mu = 1$, it is called single ton



* Normal and Subnormal fuzzy sets

Normal at least one element with $\mu = 1$

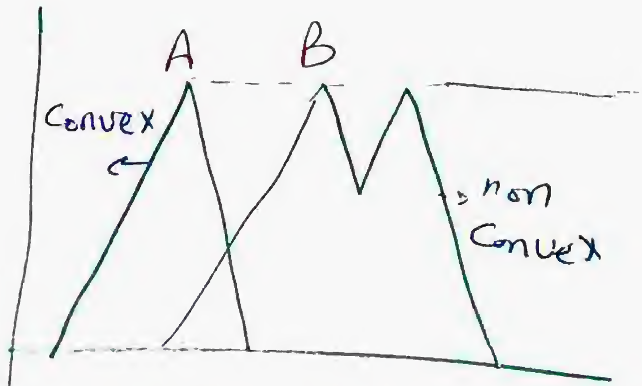
subnormal → no element with $\mu = 1$



Convex, non Convex

Convex

~~range must be from 0 → 1~~



if μ is increase or decrease or increase and ~~or~~ decrease over elements of set.

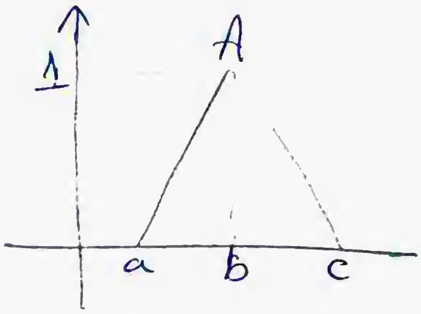
* we need fuzzy set to be : (in design)

1) Normal. 2) Convex
3) has bounded support

Fuzzy sets

1) Triangular

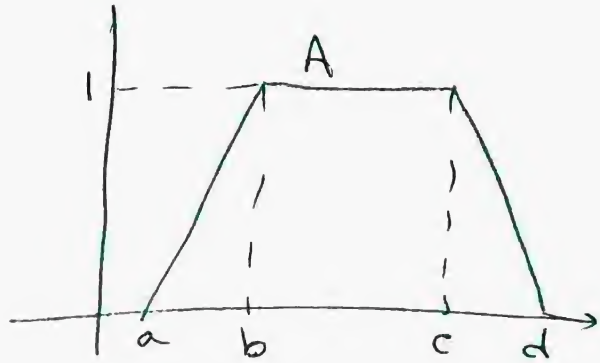
$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a} & a \le x \le b \\[2mm] \dfrac{x-c}{b-c} & b \le x \le c \\[2mm] 0 & ow \end{cases}$$



$$\mu_A(x) = \max\left[\min\left(\frac{x-a}{b-a}, \frac{x-c}{b-c}\right), 0\right]$$

2) Trapezoidal

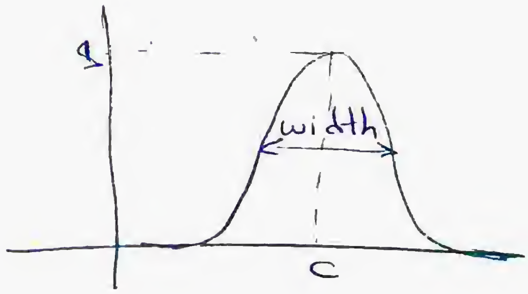
$$\mu_A(x) = \begin{cases} \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \dfrac{x-d}{c-d} & c \leq x \leq d \\ 0 & \text{otherwise} \end{cases}$$



$$\mu_A(x) = \max\left[\min\left(\frac{x-a}{b-a}, 1, \frac{x-d}{c-d}\right), 0\right]$$

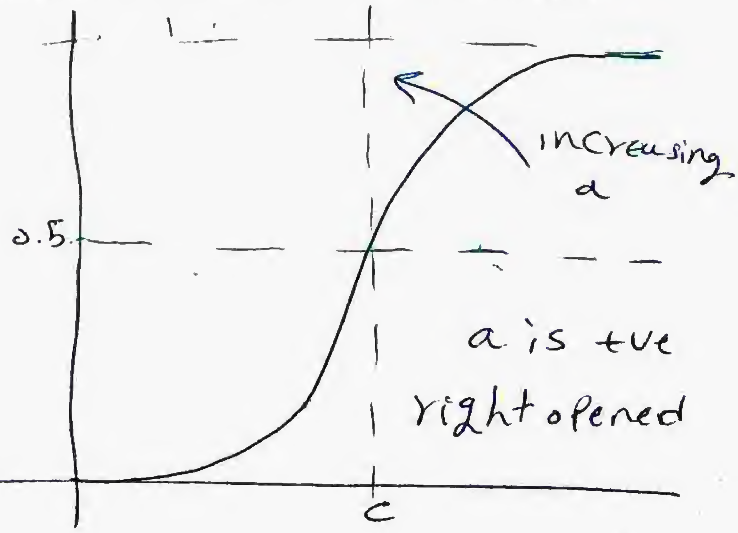
3) Gaussian

$$\mu_A(x) = e^{-0.5\left(\frac{x-c}{w}\right)^2}$$



4) Sigmoidal

$$\mu_A(x) = \frac{1}{1 + e^{-a(x-c)}}$$



Operations

Union operation

- Max operation
- Product $\mu_A + \mu_B - \mu_A \cdot \mu_B$

Intersection

- minimum $\min(\mu_A, \mu_B)$
- ~~Max~~ Product $\mu_A \cdot \mu_B$

Complement

$$\mu_{\bar{A}} = 1 - \mu_A$$

Fuzzy

slides تفريغ ال

Advantages of fuzzy controllers:-

1) cheap in cost. 2) more robustness

3) Customizable. 4) easy to design & implement

Fuzzy Apps

*washing machine.

*Rice Cookers.

* microwave ~~Microwave~~ ovens.

*Vacuum cleaners.

} Consumer Product

---

* Elevators

* Cranes

* train

* traffic control.

} → systems

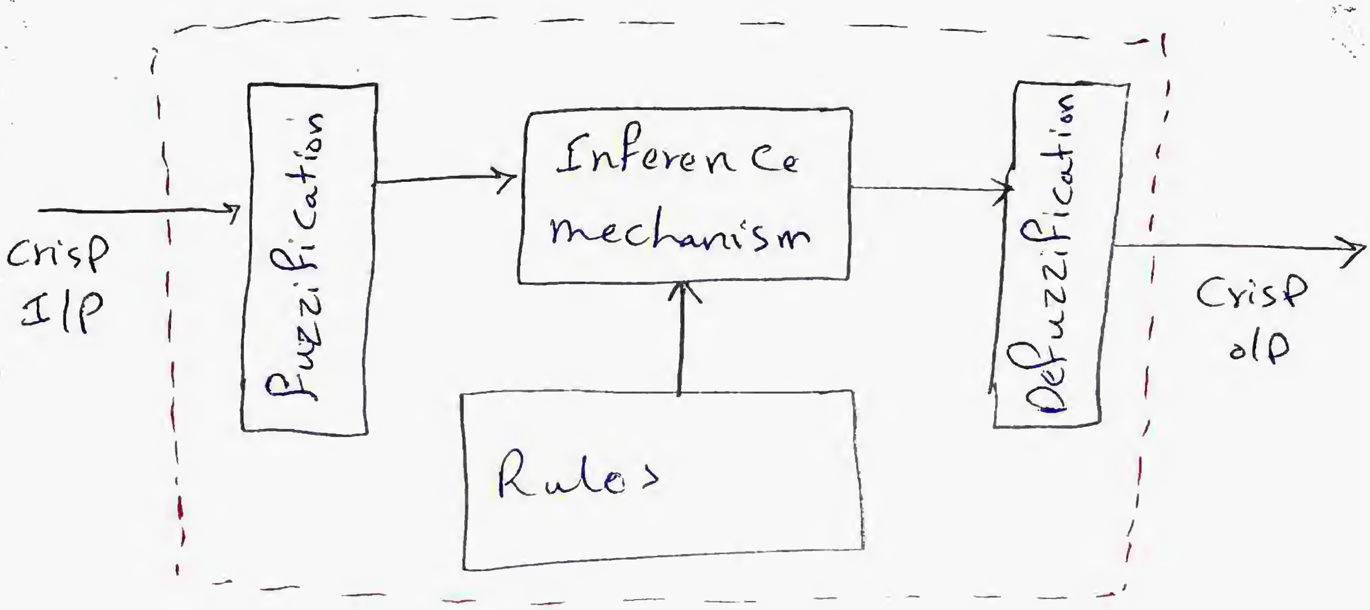
---

*Difference between fuzzy & classical (binary) sets

→ binary set: every element is member or a non-member of the set.

→ fuzzy set: every element can be member of ~~set~~ some degree and at the same time non-member to some degree at the same set.

$$\mu_A(x) \in [0, 1]$$



Fuzzy Controller

1 Fuzzification

→ Process of converting crisp values of fuzzy controller inputs into a fuzzy input sets.

→ ~~It's output is~~

2 Rules & inference mechanism

→ Rules : set of if-then statements including expert's linguistic description that governs the performance of controller.

→ Inference mechanism (heart of fuzzy control) emulates the expert's decision making in interpreting and applying knowledge about best to control plant.

→ ~~Introduction~~

③ Defuzzification

→ inverse process of fuzzification (convert fuzzy quantity into crisp value)

Rules

if ~~=~~ part ⇒ contains no. of existed potentials

then part ⇒ " the actions

* Design steps of fuzzy controller (FC)

1) Determine inputs and outputs of fuzzy controller. and desired input of system.

2) choose shapes and universe of discourses of fuzzy sets for inputs and output of FC.

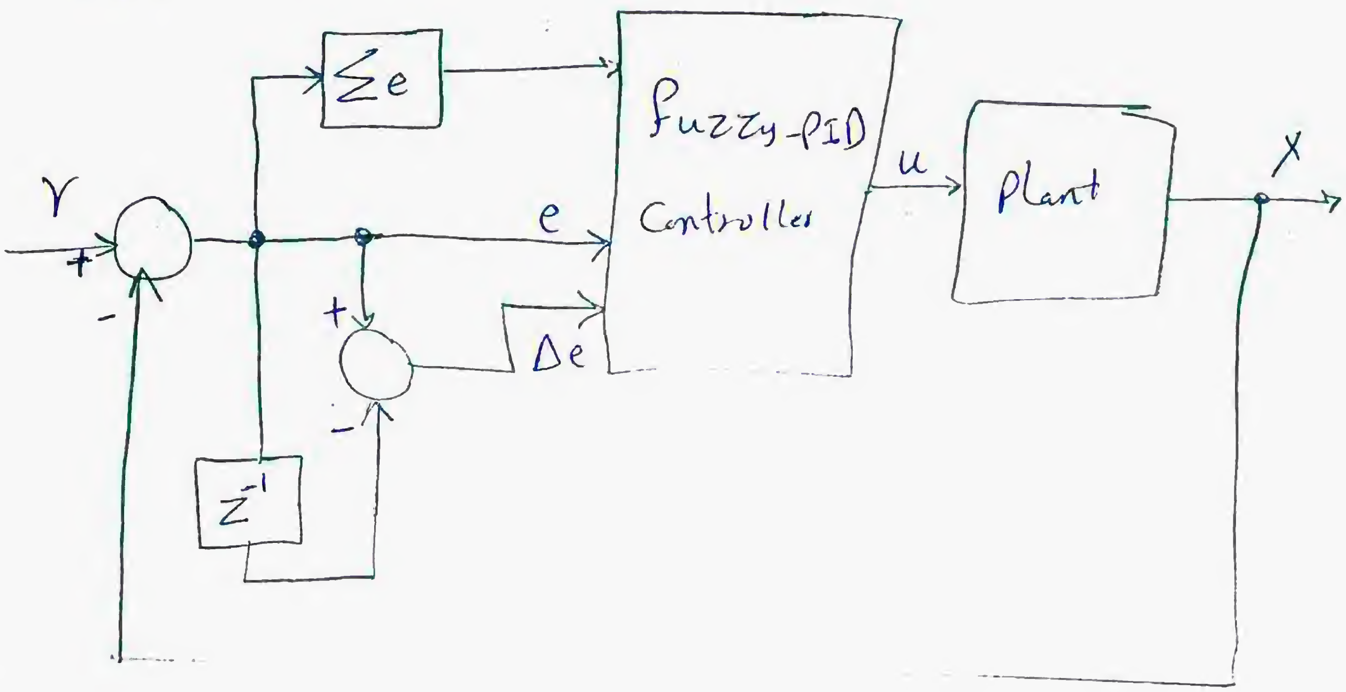
3) write suitable rules.

Fuzzy ~~PID~~ PID Controller

→ used to obtain better performance.

→ basic idea of ~~PID~~ it to chose control law by considering error e, change of error Δe and integral of error

$$u_{PID} = K_P \cdot e + K_D \cdot \Delta e + K_I \cdot \int_0^t e \cdot dt$$



* choosing shapes of fuzzy sets I/P & o/P for fuzzy controller :-

1) use normal fuzzy sets.
2) use symmetrical triangular fuzzy sets with 50 % overlap.
3) It is prefer to choose odd number of fuzzy sets (3, 5, 7, ...)
4) To ensure that universe of discourse covers all possible values of i/Ps for fuzzy controller.

Inference: Process of formulating a nonlinear mapping from given crisp i/p to crisp output.

Types of fuzzy inference

1) Mamdani fuzzy inference
2) Sugeno " "
3) Tsukamoto " "

$\Big\}$ most commonly

ـــ التحويل يكون (non linear) لأنه مستحيل العمليات دي
كلها تكون (linear).

$\boxed{1}$ Mamdani

→ rules obtained from an experienced human operator

ex

R1: if X_1 is A_1 and X_2 is B_1 then y is $C1$

output of each rule is the truncated membership functions from minimum firing strength.

methods of defuzzification (in mamdani)

a) center of gravity (CoG)

b) weighted average method

c) Mean-max membership method

2 Sugeno or TSK fuzzy inference

rules linear function which is combination of input variables plus constant term.

EX

R1: if x_1 is A_1 And x_2 is B_1 then $Y_1 = P_1 x_1 + q_1 x_2 + r_1$

deffuzzification (weighted average)

3 Tsukamoto fuzzy inference

rules

↳ It is monotic MF

ex

R_1: if x_1 is A_1 and x_2 is B_1 then y is C_1

defuzzification

weighted average.

main difference between 3 ways

1) Rules (then Part)

and write rules of each way.

2) way of apply deffuzzification.

---

* لو ال (o/p) بسبب ال (fuzzy) يروح لل

(steady state) بس بالسالب فيه ٣ حلول :-

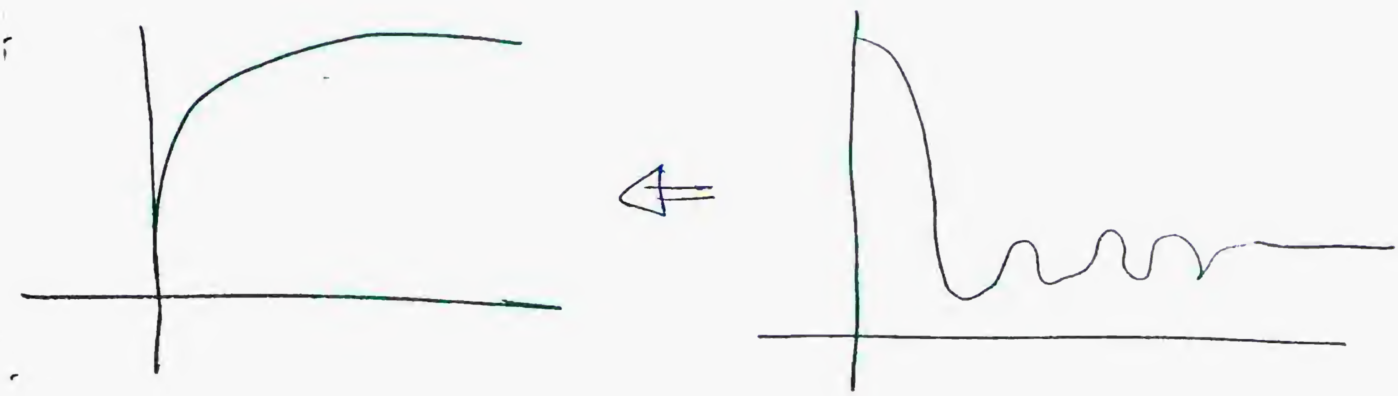
١) تغير ال (gain) للسالب .

2) تعكس ال (rules) ⇐ عند جمع ال (rules)

مش هحط عكس المجموع الجبرى.

3) اعكس مفهوم ال (error)

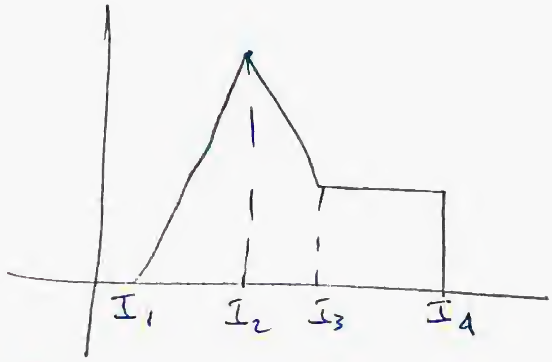
$e = r - y \Rightarrow e = y - r$



ways of defuzzifications.

Center of gravity

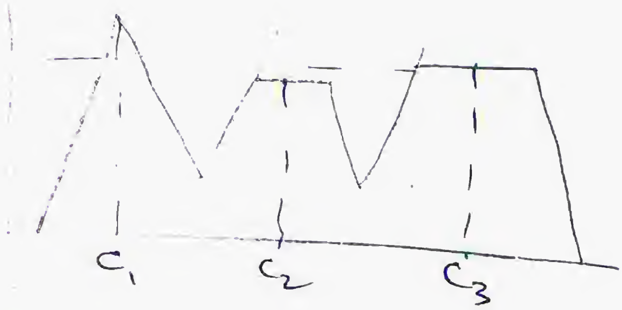
$$U^{crisp} = \frac{\int \mu(u)\, u\, du}{\int \mu(u)\, du}$$



← كل فترة وتكاملها

~~Mean~~ Max - mean

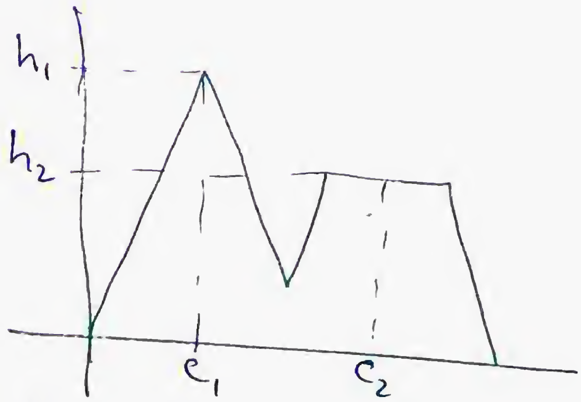
$$U^{crisp} = \frac{c_1 + c_2 + c_3}{3}$$



القيم المناظرة لل maximum

weighted average

$$U^{crisp} = \frac{c_1 h_1 + c_2 h_2}{h_1 + h_2}$$



لو فيه تماثل تستخدمها.

Fuzzy-PID Controller

↳ is used to obtain better performance in respect of rise time, settling time, overshoot and steady-state error.

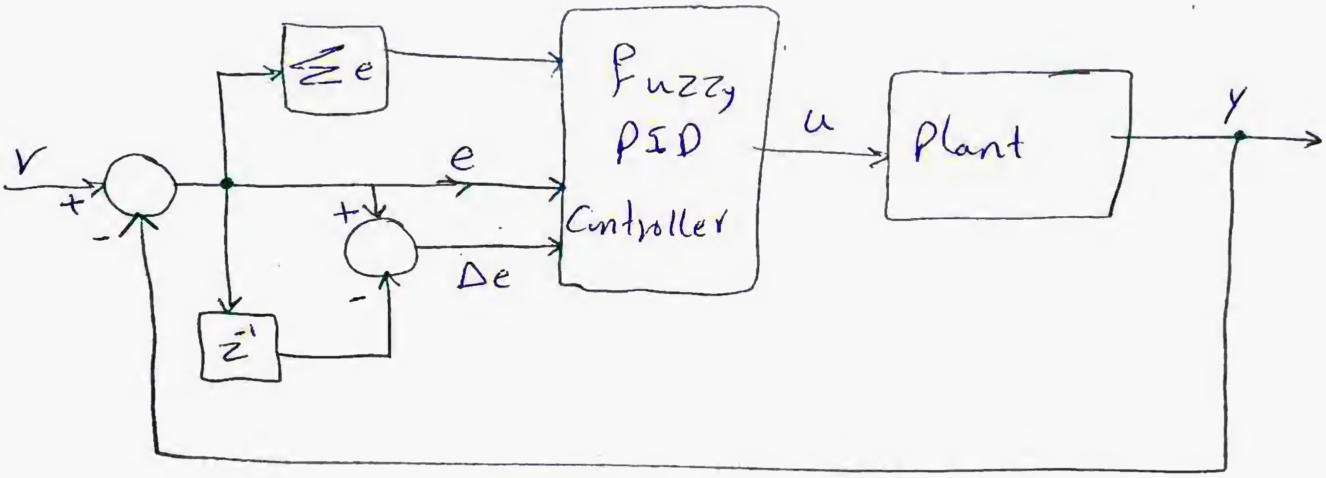
Basic idea of PID

↳ to choose control low by considering error, ~~and~~ change of error and integral of error.

$$u_{PID} = K_p \cdot e + K_D \cdot \Delta e + K_i \int_0^t e\,dt$$

For discrete:

$$u_{PID} = K_p \cdot e + K_D \cdot \Delta e + K_i \sum e$$



→ الـ system) ده مش مناسب من الناحية العملية لأن عندك
(3 inputs) يعني (huge no. of rules) يعني
(Fast response) في حين ان التطبيقات محتاجة (long reasoning time)

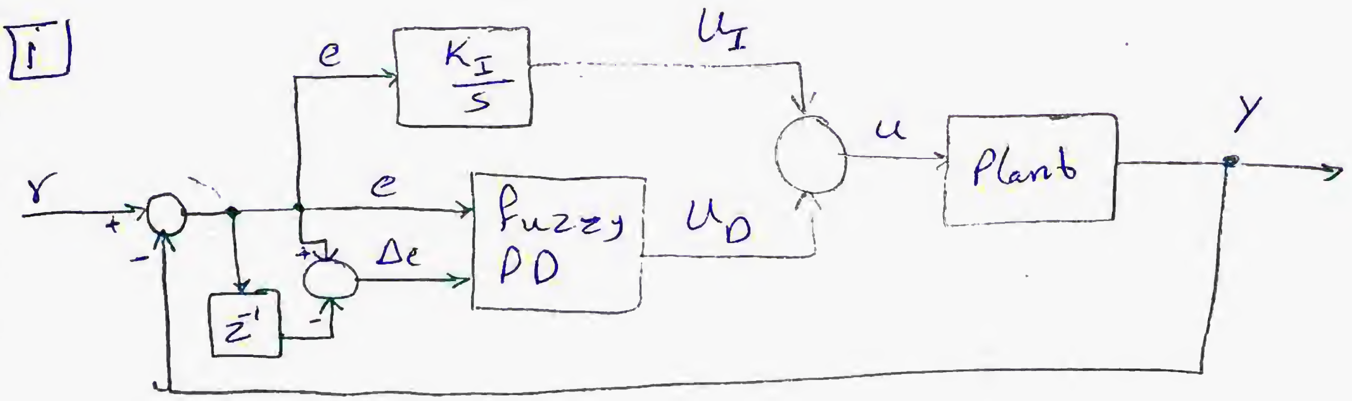
→ There are various methods used to ~~def~~ design fuzzy-PID Controller, but the two most common methods:

1) Combination fuzzy-~~PID~~ PD Controller with conventional integral controller.

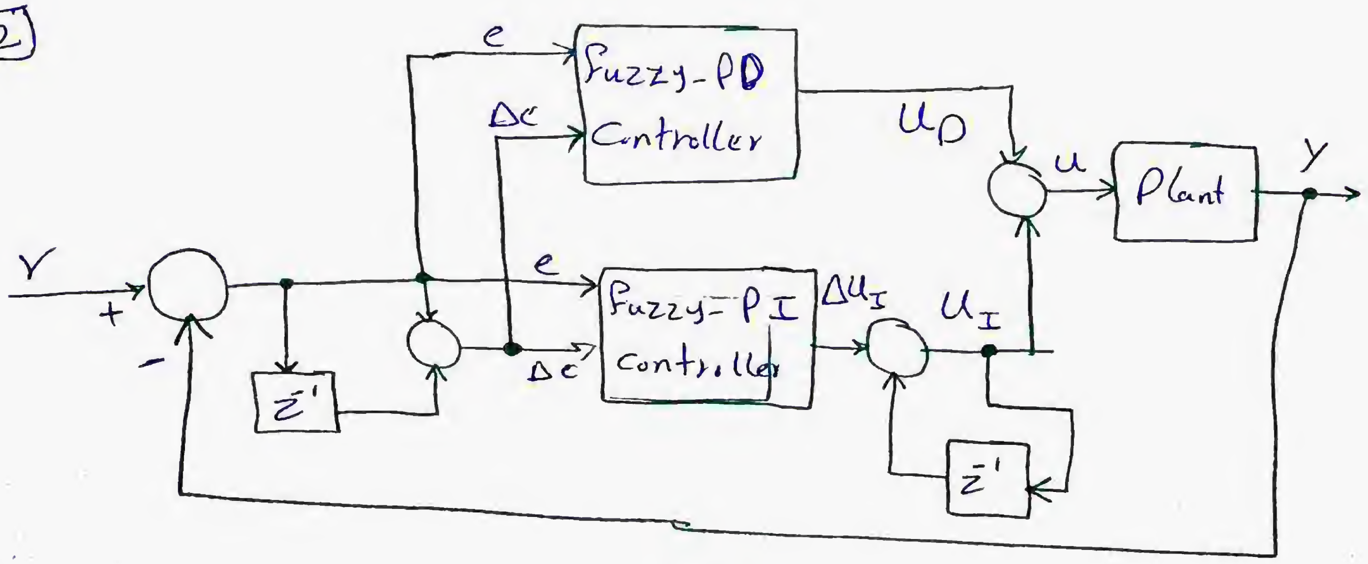
2) Combination of fuzzy-PD controller with fuzzy-PI controller.

⇒ In conventional integral controller (gain K_i is to be determined by trial and error)

1



2



ANFIS

Evolutionary Computation

↳ Heuristic algorithms based on Principles of Darwinian evolution

Example of these algorithms:-

a) genetic algorithm.
b) Differential evolution algorithm.
c) Fish swarm.
d) Artifial Bee Colony.
e) Particle swarm.

ANFIS based on TSK fuzzy inference system

↳ AnFIS is fuzzy system modelled in the form of the artifial neural network. ~~so that~~

⇒ why ANFIS is based on TSK fuzzy inference?

Cause ~~fuzzy~~ TSK is simple in computation and easy to be combined with optimizing and self-adapting methods.

ANFIS stands for:
↳ Adaptive neuro fuzzy inference system.
↳ Adaptive network-based fuzzy inference system.

* What is the main objective of ANFIS?

↳ It is to determine the optimum values of the equivalent fuzzy inference system parameters (of TSK type) by applying a learning algorithm using input-output datasets (train and test)

* What are the parameters to be obtimized in ANFIS?

1) Premise (if part) which describe shape of MF_s.

2) Consequent (then part) " " overall output of system.

* Explain with example the type of methods that used to optimize ANFIS parameters?

1) derivative - based methods

↳ back propagation (BP) ↳ least squares estimate (LSE)

↳ hybrid learning (HL) Combination of LSE & BP

2) derivative - free methods

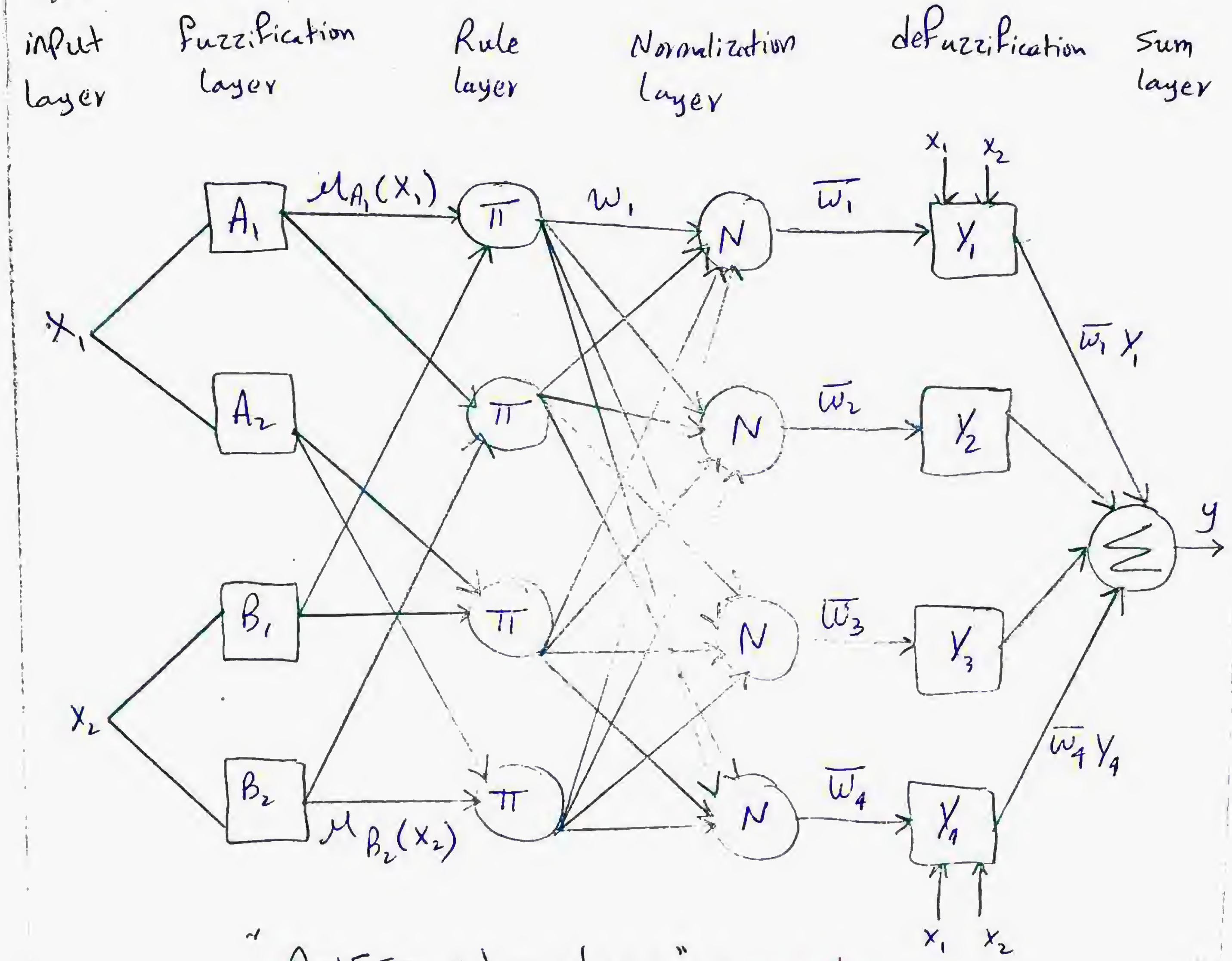
↳ genetic algorithm (GA)

↳ particle swarm optimization (PSO)

↳ differential evolution (DE)

↳ shuffled frog leaping Algorithm (SFLA)

↳ artifial bee colony algorithm (ABC)



"ANFIS structure" 2 inputs

Layer 0: input layer

↳ as inputs are applying to system.

Layer 1: fuzzification layer

↳ applying inputs MFs and produce a ~~measure of~~ degree of membership. (μ)

Layer (2): Rule layer

↳ its output represents fire strength of rule.

↳ executes fuzzy of antecedent (if) part

Layer 3: normalization layer

↳ output is the ratio of firing strength of ith rule to sum of all firing strengths rule.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4} \quad , i = 1, 2, 3, 4$$

Layer 4: defuzzification layer:-

↳ it executes consequent part of fuzzy rules

↳ its output is product of normalized firing strength rule & its corresponding linear function in consequent part.

Layer 5: Sum layer:-

↳ computes total crisp output of fuzzy system.

$$y^{crisp} = \frac{w_1 Y_1 + w_2 Y_2 + w_3 Y_3 + w_4 Y_4}{w_1 + w_2 + w_3 + w_4} = \bar{w}_1 Y_1 + \bar{w}_2 Y_2 + \bar{w}_3 Y_3 + \bar{w}_4 Y_4$$

Notes

* no. of premise parameters = no. of control parameters of MFs * total no. of MFs

* no. of consequent parameters = (no. of inputs + 1) * no. of rules

* total no. of ANFIS parameters =

no. of premise parameters + no. of consequent parameters

optimization   Lec 8, 9

Meaning of optimization:-

↳ minimize or maximize of certain objective function.

Aim of optimization

↳ find best (optimum) solution for any optimization problem.

→ **Examples of optimization Problems / APPs...**

a) optimize Parameters of ANN model (weights & bias)

b) " Parameters of ANFIS model (Premise & antecedent parameters)

c) Tuning Parameters of PID (get best value of K_P, K_I & K_D)

d) Getting best Placement of WI-FI access point for Indoor positioning system (IPS)

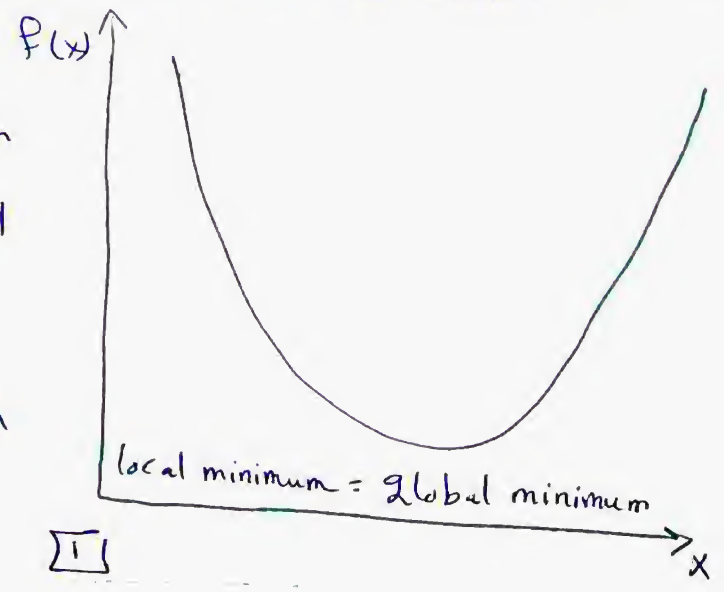
Notes

لـ كل (global optimum) هو (local optimum) لكن العكس مش دايماً صح.

* **Unimodel function**

↳ has single local minimum which is itself the global optimum.

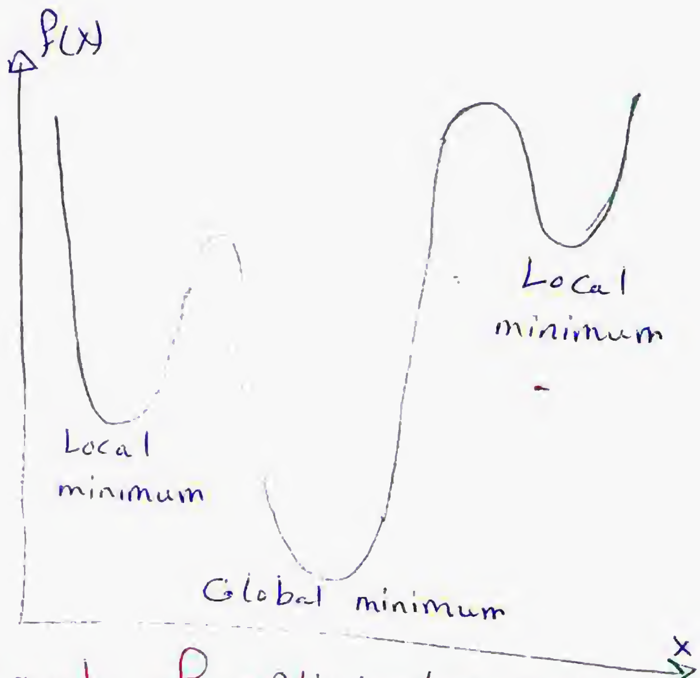
f(x) → objective function to be minimized.



* The ~~local~~ global minimum is the least among all local minimum.

2 Multimodel Function:-

→ has more than one local optimum and one global optimum ~~functions~~



* what is the ideal target of optimization problem?
↳ It is the global optimum (a good optimization algorithm does not get trapped in any local optimum)

Basic elements of optimization Problem

1) An objective function f
↳ function to be optimized (minimized or maximized)

2) The number of components or variables of the objective function that specifies the dimensionality of the optimization problem

$$f(x_1, x_2, \dots, x_D)$$

where D is no. of variables specifies dimensionality of the problem.

$f(x)$, $X = [x_1, x_2, \dots, x_D]$ $1 \times D$ vector

3) sets of constraints forced on required solution
↳ most Problems Constrain at least search domains of the variables vector $X = [x_1, x_2, \dots, x_D]$
↳ aim of optimization is to find global optimum $X^* \subseteq R^D$ from allowable search domains, where $f(x^*)$ has the minimum value in search domain.

Classification of optimization Problems

Classification basis	Types of optimization Problem	
Dimensionality (D)	univariate (D = 1)	Multivariate (D > 1)
Linearity	Linear	Non Linear
Constraints	unConstrained (only search ranges of x_d are Constrained)	Constrainted (Additional Constraints are Forced on x_d)
no. of optimum values	unimodel (one optimum only)	Multimodel
no. of objective	single-objective	Multi-objective (more than one objective to be optimized)

Separability of Variables X_d	Separable Function of $f(x_1, \dots, x_D)$ can be divided to D functions in form: $f(x_1) + f(x_2) \dots + f(x_D)$	Non-separable Can't be divided.

Evolutionary optimization Algorithms
(Population-based " ")

↳ Evolutionary optimization algorithms are Population based of candidate solutions, not Just one solution.

what is the basic characteristic of Population-based

↳ the iteration Policy depends on a Population.

what happens during the iteration?

↳ Population of constant size is maintained, and group of solutions is improved Progressively.

Note that "can be neglected"

↳ Having group of solutions "working together is the Key of emulating behavior of biological organisms in modern biology-inspired optimization approaches (e.g. flock of birds, school of fish)

Examples of Evolutionary optimization Algorithm

1) Genetic algorithm (GA)
2) Bat algorithm (BA)
3) Artificial Bee colony (ABC)
4) Differential evolution (DE)
5) Ant colony optimization (ACO)
6) Particle swarm optimization (PSO)

Exploration & Exploitation

	Exploration	Exploitation
Meaning	↳ Find new solutions in search domains which haven't been evaluated before.	↳ try to improve the current found solution by performing small changes that lead to new solutions.
Variation of population members from one iteration to another	large	very small

Basic element affect on Exploration & Exploitation

1) Population size (no. of members in population) affects on exploration rate.

Large size of population ⟹ ↑↑ rate of exploration.

2) Control Parameters of optimization algorithm
↳ affect on exploration and exploitation.

Notes

* optimization algorithm starts from larger exploration rate
↳ this allows the algorithm to cover large regions of search domains quickly.
* As iterations processes: exploration rate decreased
↳ allows to exploit the promising regions that are previously explored.

Benchmark Functions

↳ standard complex mathematical functions with different ch/s are used to test optimization algorithm "to evaluate efficiency & robustness"

How this test happens?

↳ بعد اختيار مجموعة الـ (benchmark functions) المناسبة ، الـ (Algorithm)
بيبدأ يـ(run) مع الدوال دي لعدد N من المرات ~~...~~. كل
عملية (run) بتحتوي على (no. of iterations)

↳ نتيجة الاختيار بتطلع رقم الـ successful runs for each fn.

⇒ run is considered successful if algorithm reached the required global optimum.

Common Benchmark Functions

Benchmark Functions		Search range	Functions Properties
Sphere Function	$f_1(x) = \sum_{i=1}^{i=D} x_i^2$	$[-100, 100]^D$	unimodel separable
Rosenbrock Function	$f_2(x) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$	$[-2.048, 2.048]$	uimodel $(D<4)$ Multimodel $(D \geqslant 4)$ nonseparable
Ackley Function	$f_3(x) = 20 + e - 20\, e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)}$	$[-30, 30]^D$	Multimodel non separable
Griewank Function	$f_4(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]^D$	Multimodel non-~~Sepa~~ separable
Rastrigin Function	$f_5(x) = \sum_{i=1}^{D} \left[10 + x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]^D$	Multimodel Separable
Schwefel Function	$f_6(x) = 418.9829\, D - \sum_{i=1}^{D} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500, 500]^D$	Multimodel separable

Note that All of these Functions are

*single-objective. *unconstrained

Lec 9

Differential Evolution (DE) optimization Algorithm

*Basics of Differential Evolution:

↳ Population-based optimization algorithm.

↳ Developed to optimize real parameter, real valued functions.

→ General problem formulation is: for an objective function $f: X \subseteq R^D \rightarrow R$ where $X \neq \phi$

↳ minimization problem is to find $x^* \in X$ such that $f(x^*) \leq f(x)\ \forall x \in X$ where $f(x^*) \neq -\infty$

↳ DE is a parallel search method which utilizes NP & D-dimensional parameter vectors.

Note

→ suppose we want to optimize a function with D-dimensional real parameters R^D

↳ we select population size NP (NP must be $\geqslant 4$)

$$X_{i,G} = [X_{1,i,G}, X_{2,i,G}, \ldots, X_{D,i,G}] \quad i = 1,2,\ldots,NP \quad J = 1,2,\ldots,D$$

where: $X_{i,G}$ → Parameter vector in population for each generation.

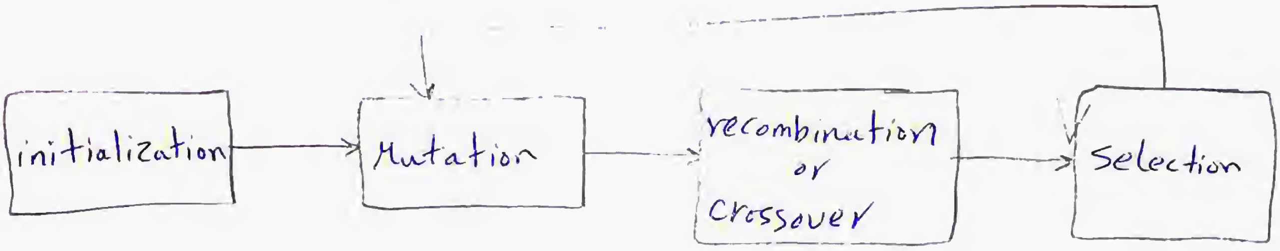
G → generation number

* NP → doesn't change during minimizing process.

Differential Evolution (DE)

→ It is an Evolutionary algorithm.

→ " " considered from initialization and cycle of stages of [mutation, recombination (or crossover) and selection]



[1] Initialization

↳ all parameter vectors in population are randomly intialized

1) تحدد لكل (Parameter) حد أقصى وأدنى

$$X_J^{L} \leq X_{J,i,1} \leq X_J^{U}$$

($L \rightarrow$ lower, $U \rightarrow$ upper)

2) بشكل عشوائي إختيار القيمة المبدائية لل (Parameter) في الحدود

$$[\, X_J^{L} \,,\, X_J^{U} \,]$$

3) كإقتراح إختر قيم عشوائية ما بين الحد الأعلى و الأصغر.

$$X_{J,i,1} = X_J^{L} + \text{rand} * (X_J^{U} - X_J^{L})$$

2 Mutation

→ Mutation, ~~se~~ recombination and selection will run for each NP parameter vectors of population.

⇒ randomly select three vectors ($X_{r1,G}$, $X_{r2,G}$, $X_{r3,G}$) where i, r_1, r_2, r_3 (distinct int.) $\in \{1, 2, \ldots, NP\}$

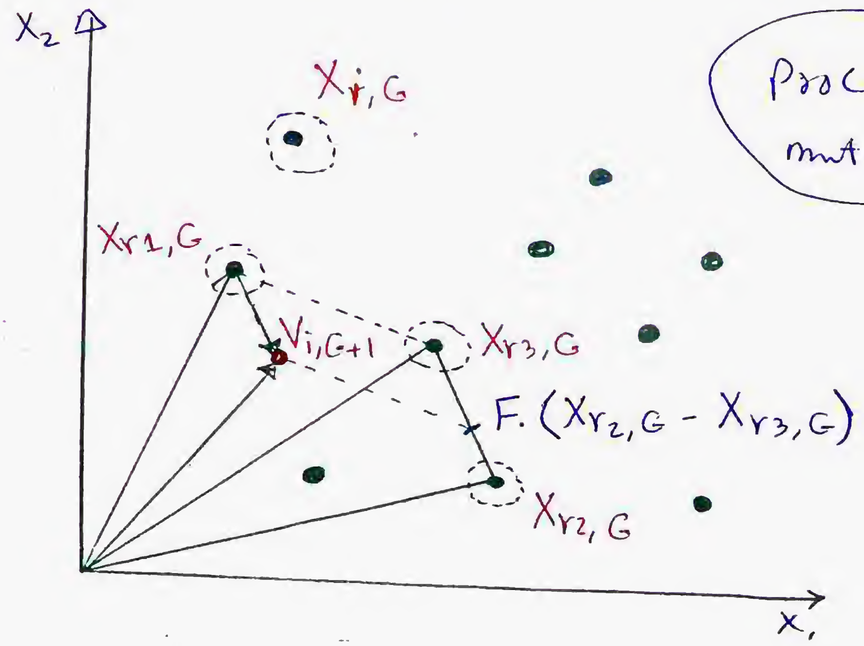
⇒ Add weighted difference of $X_{r2,G}$ & $X_{r3,G}$ to the base vector $X_{r1,G}$

$$V_{i,G+1} \; \cancel{V_{i,G+1}} = X_{r1,G} + F.\,(X_{r2,G} - X_{r3,G})$$

where : F → mutation factor [0,2)

↳ controls the amplification of differential variation $(X_{r2,G} - X_{r3,G})$

$V_{i,G+1}$ → mutant vector or donor vector.



Process of generate mutant vector

3 Crossover

لم يتم دمج الـ(successiful solutions) من العملية السابقة.

* trial vector $U_{i,G+1}$ developed from
  - elements of target vector $X_{i,G}$.
  - elements of mutant vector $V_{i,G+1}$

$$U_{J,i,G+1} = \begin{cases} V_{J,i,G+1} & \text{if } (\text{randb}(J) \leq CR) \text{ or } J = \text{Irand} \\ X_{J,i,G} & \text{if } (\text{randb}(J) > CR) \text{ or } J \neq \text{Irand} \end{cases}$$

$$i = 1, 2, \dots, NP \quad \& \quad J = 1, 2, \dots, D$$

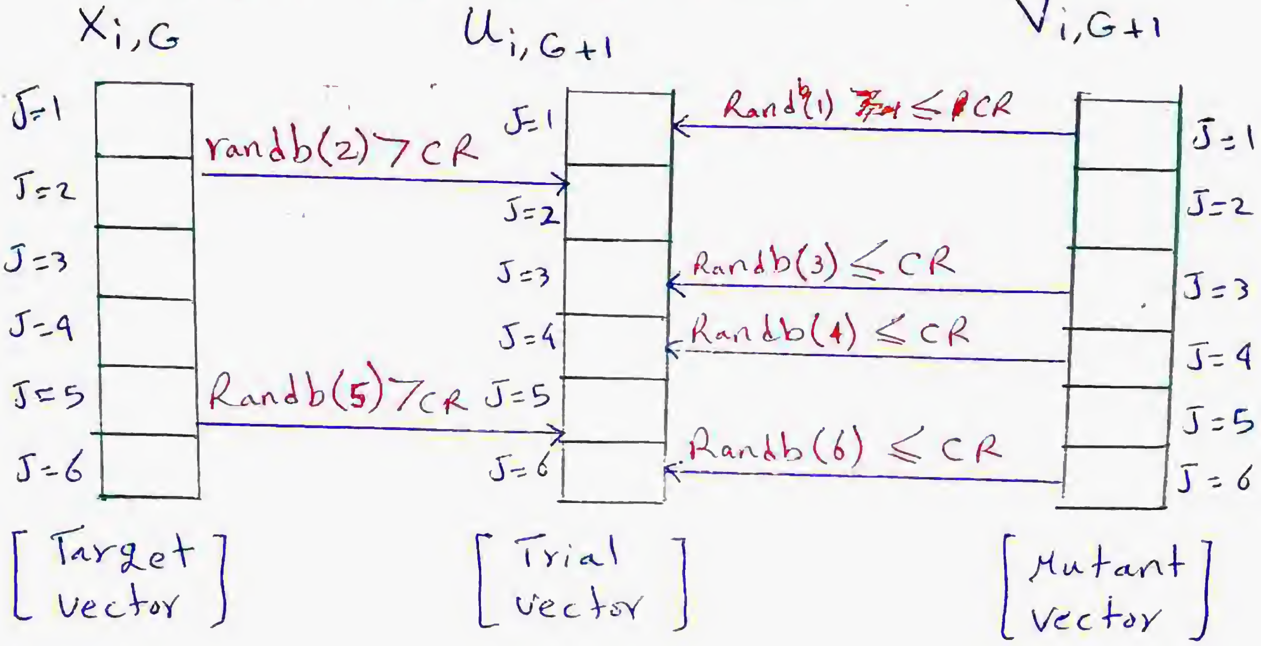
where:

randb(J) → is the J^{th} evaluation of a uniform random number generator with out come $\in [0, 1]$

CR → "crossover rate" & it is constant $\in [0, 1]$
  → determined by user.

Irand → random integer from $[1, 2, \dots, D]$ which ensures that the trial vector $U_{i,G+1}$ gets at least one parameter from $V_{i,G+1}$.

Crossover for $D = 6$ Parameters.



<u>Notes</u>

↳ After crossover process, some or all components of the trial vectors may be lie outside search domain.

↳ to be sure that these components are within predifned constraints we use

$$U_{J,i,G+1} = \begin{cases} X_J^U + rand_{J,i} \cdot (X_{J,i,G} - X_J^U), & \text{if } (U_{J,i,G+1} > X_J^U) \\ X_J^L + \quad " & \text{if } (U_{J,i,G+1} < X_J^L) \end{cases}$$

<u>or</u>

$$U_{J,i,G+1} = \begin{cases} 2X_J^U - U_{J,i,G+1} & \text{if } (U_{J,i,G+1} > X_J^U) \\ 2X_J^L - U_{J,i,G+1} & \text{if } (U_{J,i,G+1} < X_J^L) \end{cases}$$

or

$$u_{J,i,G+1} = \begin{cases} (X_J^U + X_{J,i,G})/2 & (\text{if } u_{J,i,G+1} > X_J^U) \\ (X_J^L + X_{J,i,G})/2 & (\text{if } u_{J,i,G+1} < X_J^L) \end{cases}$$

$\boxed{4}$ selection

لم هنقارن (trial vector) $u_{i,G+1}$ بال (target vector) $X_{i,G}$
وهنختار اللي (cost function) بتاعته أقل.

$$X_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } [f(u_{i,G+1}) \leq f(X_{i,G})] \\ X_{i,G}, & \text{otherwise} \end{cases}$$

Note that

↳ Mutation, crossover and selection continue until some stopping criterion is reached.

Meaning of name [DE/rand/1/bin]

DE → differential evolution

rand → ~~best~~ base vector for mutation is chosen randomly.

1 → one difference vector is used to construct the donor.

bin → crossover is binomial.

*Mutant vector $V_{i,G+1}$ is generated according to one of the following eqns..

(1) $V_{i,G+1} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G})$ DE/rand/1/bin

(2) $V_{i,G+1} = X_{best,G} + F.(X_{r1,G} - X_{r2,G})$ DE/best/1/bin

3) $V_{i,G+1} = X_{i,G} + F.(X_{best,G} - X_{i,G}) + F.(X_{r1,G} - X_{r2,G})$

DE/target-to-best/1/bin

4) $V_{i,G+1} = X_{best,G} + F.(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G})$

DE/best/2/bin

5) $V_{i,G+1} = X_{r1,G} + F.(X_{r1,G} - X_{r2,G}) + F.(X_{r3,G} - X_{r4,G})$

DE/best/2/bin

DE Control Parameters

control Performance of DE

1) Population size (NP) 2) mutation factor (F)

3) crossover rate (CR)

↳ These Parameters chosen carefully to avoid the state of stagnation (or Premature convergence) for the DE algorithm.

* Role of NP

↳ affects ability to search the parameter space.

→ why NP must be ≥ 4?

↳ cause small values of NP result in few numbers of mutant vectors that may cause insufficient exploration (premature convergence)

↳ but large values cause excessive exploration (slow convergence) and increase no. of computations.

NP = 30 → small dimension values $D < 30$

NP = 30 → large dimension values $D \geq 30$

Mutation Factor (F)

↳ is relevent to convergence speed ~~(as it~~ (as it responsible for step size that interferes in the formation of mutant vector)

↳ small values of F ⇒ Premature convergence.

↳ F > 1 (large steps) ⇒ slow convergence.

⇒ good initial choise is F = 0.5 for effective range [0.4, 1]

Cross-over rate (CR)

↳ Controls no. of changes in Parameters of a Population member.

Small value of CR (0 or 0.1) (strong crossover) ⇒ lead to:

↳ most changes occurring along one dimension or small subset of dimensions, this is useful for separable functions.

Large values (near to 1)

↳ most Components being chosen from mutant vector, suitable for non-separable.

CR → [0, 0.2] → separable

CR → [0.9, 1] → non-separable.